WHAT IS CLAIMED IS:

1. A method for creating and/or modifying a dynamically update-able, searchable packet classification databank, comprising the steps of:

receiving a collection of packet classification rules, each packet classification rule being represented as a plurality of binary locations;

selecting an index key based on a common location among said packet classification rules at a first level, such as to enable partitioning of said collection into two or more siblings at a second level, wherein the binary value of said common location represents a feature whereby the composition of each sibling contains packet classification rules possessing a common feature; and

selecting an index key based on a second common location among said packet classification rules at said second level, such as to enable partitioning of at least one of said two or more siblings at said second level into two or more siblings at a third level.

2. The method of claim 1, further comprising the step of:

selecting an index key based on a third common location among said packet classification rules at said third level, whereas to enable partitioning of at least one of said two or more siblings at said third level into two or more siblings at a fourth level.

3. The method of claim 1, further comprising the step of:

repetitively partitioning each sibling at a respective level into two or more siblings at a lower level until reaching a partition threshold.

4. The method of claim 3, wherein said partition threshold is predicated on a maximum number of rules residing in said sibling at said respective level.

5. The method of claim 3, wherein said partition threshold is predicated on a maximum number of levels.

6. The method of claim 1, wherein each sibling at a respective level has a substantially equivalent quantity of said packet classification rules.

7. The method of claim 1, wherein each of said selecting an index key step comprises the steps of:

measuring a difference in cardinality at each location coordinate that has not been selected previously as an index key; and

computing an optimization parameter for each location coordinate.

8. The method of claim 7, wherein each of said selecting an index key step further comprises the step of:

selecting an index key corresponding to a location coordinate having an optimization parameter closest to a predetermined value.

9. The method of claim 7, wherein each of said selecting an index key step further comprises the step of:

selecting an index key corresponding to a first location coordinate determined to have an optimization parameter closest to a predetermined value in response to determining multiple location coordinates having an optimization parameter closest to a predetermined value.

10. The method of claim 1, further comprising the step of:

receiving at least one packet classification rule within said collection that has one or more location coordinates denoted as both binary values.

11. The method of claim 10, wherein each of said selecting an index key step comprises the steps of:

measuring a difference in cardinality at each location coordinate that has not been selected previously as an index key; and

computing an optimization parameter for each location coordinate.

12. The method of claim 11, wherein said computing an optimization parameter comprises:

determining an evenness of division for siblings at a respective level; and

determining an average cardinality.

13. The method of claim 1, further comprising the steps of:

receiving at least one packet classification rule within said collection that has two or more location coordinates that denote a feature having a range of values; and

decomposing said at least one packet classification rule into two or more packet classification divisional rules, wherein said selecting an index key steps include processing said divisional rules as part of said collection.

14. The method of claim 1, further comprising the step of:

manifesting a query key based on index keys selected to partition said packet classification rules.

15. The method of claim 14, further comprising the steps of:

enabling addition and/or deletion of a packet classification rule in said collection; and

revising said query key in response to said addition and/or deletion of a packet classification rule.

16. The method of claim 15, further comprising the step of:

performing said revising said query key on a periodically scheduled basis.

17. The method of claim 15, further comprising the step of:

performing said revising said query key on demand.

18. The method of claim 14, further comprising the steps of:

receiving a packet;

applying said query key to said packet to produce a packet key; and

searching said collection to detect a packet classification rule matching said packet key.

19. The method of claim 18, further comprising the steps of:

detecting multiple packet classification rules matching said packet key; and

selecting a collision location key based on a common location to enable partitioning of said multiple packet classification rules.

20. The method of claim 18, further comprising the steps of:

detecting multiple packet classification rules matching said packet key; and

sequentially comparing each of said multiple packet classification rules with said packet to detect a matching rule

21. The method of claim 18, further comprising the step of:

enabling addition and/or deletion of a packet classification rule in said collection during said searching said collection.

22. A packet classification system, comprising:

a first memory for receiving a collection of packet classification rules, wherein each packet classification rule is represented as a plurality of binary locations; and

a mask constructor for selecting one or more index keys,

wherein each index key is based on a common location among said packet classification rules residing at a level, and enables partitioning of said packet classification rules into two or more siblings at another level, and

wherein said mask constructor continues to select index keys to repetitively partition each sibling at a respective level into two or more siblings at a lower level until reaching a partition threshold.

23. The system of claim 22, wherein said mask constructor assembles said one or more index keys into a query key.

24. The system of claim 23, further comprising:

a key extractor for applying said query key to produce a refined rule collection from said collection located within said first memory; and

a second memory for storing said refined rule collection.

25. The system of claim 24, wherein said second memory is a content addressable memory.

26. The system of claim 23, further comprising:

a key extractor for applying said query key to an incoming packet to produce a packet key.

27. The system of claim 26, further comprising:

a packet classifier for applying said packet key to detect a packet classification rule matching said packet key.

28. The system of claim 26, wherein said key extractor is a multiplexor, wherein said multiplexor is configured to select field descriptors from said packet based on said query key.

29. The system of claim 28, wherein said multiplexor is a crossbar switch or a bit shifter.

30. A computer program product comprising a computer useable medium having computer readable program code means embedded in said medium for causing an application program to execute on a computer used to classify packet flows, said computer readable program code means comprising:

a first computer readable program code means for causing the computer to select one or more index keys,

wherein said first computer readable program code means selects each index key such that each index key is based on a common location among a set of packet classification rules residing at a level, and enables partitioning of said set into two or more siblings at another level, and

wherein said first computer readable program code means continues to select index keys to repetitively partition each sibling at a respective level into two or more siblings at a lower level until reaching a partition threshold; and

a second computer readable program code means for causing the computer to assemble said one or more index keys into a query key.